

ELECTRONIC CONTROLLER

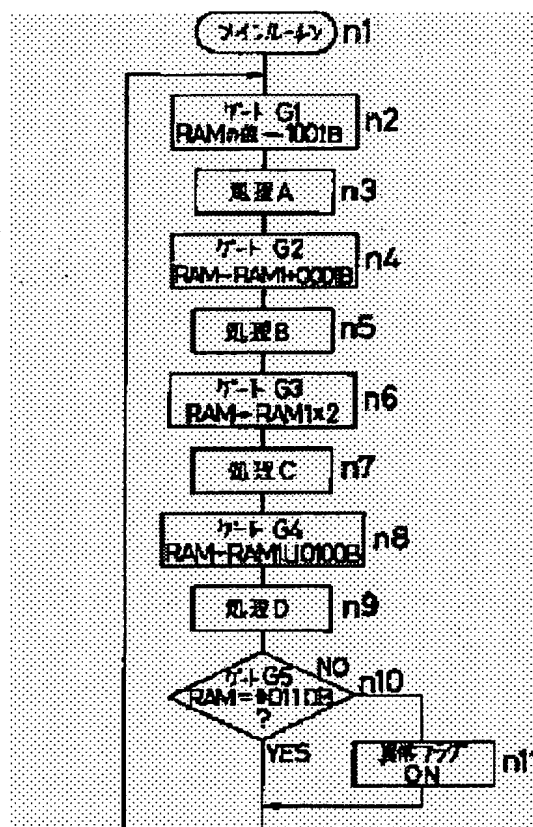
Patent number: JP6149626
Publication date: 1994-05-31
Inventor: SASAKI YOSHIHIRO
Applicant: FUJITSU TEN LTD
Classification:
- international: G06F11/30; G06F11/00
- european:
Application number: JP19920303927 19921113
Priority number(s):

031356 U.S. PTO
10/759070



Abstract of JP6149626

PURPOSE: To improve the reliability of an electronic controller by checking the functions of a CPU used for the control device, an ALU used in the CPU and a register part.
CONSTITUTION: Arithmetic processing is executed in each plural gate processing inserted into a main routine to be executed by the CPU and processing results are successively stored (n2, n4, n6, n8). In the final gate, the stored values are compared with an expected value (n10, n11) to check the validity of the CPU, the ALU and the register part.



THIS PAGE BLANK (USPTO)

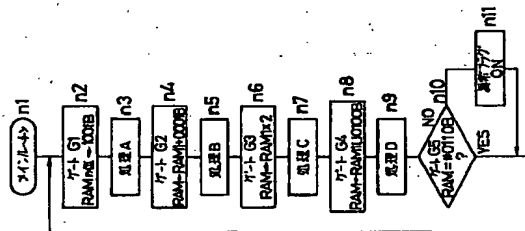
(19)日本特許庁(JP) (12)公開特許公報(A) (11)特許出願公開番号
特開平6-149626
(43)公開日 平成6年(1994)5月31日

(51)IntCl. G 0 6 F 11/20 3 0 5 J 9230-5B 11/00 3 5 0 E 7313-5B	FI 特許庁表示箇所
(21)出願番号 特願平4-30327	(71)出願人 00023752 富士通テン株式会社 兵庫県神戸市兵庫区御所通1丁目2番28号 佐々木 雄弘
(22)出願日 平成4年(1992)11月13日	(72)発明者 神戸市兵庫区御所通1丁目2番28号 富士通テン株式会社内 井 理士 西 敏 圭 一郎
(23)発明の名称 電子制御装置	(74)代理人 井 理士 西 敏 圭 一郎

(57)【要約】

【目的】 電子制御装置に用いられているCPUと、そ
の中で使用されるALUおよびレジスタ部の機能をチェ
ックし、信頼性の向上を図る。

【構成】 CPUが実行するメインルーチンに導入され
た複数のゲート処理ごとに、メインルーチンの処理手順
に従って演算処理、処理結果を順次記憶させる(n
2、n4、n8、n10)。最終ゲートにおいて前記記憶
された値と、期待値とを比較する(n10、n11)こ
とにより、前記CPUをはじめALU、レジスタ部の機
能の良否をチェックする。



【特許請求の範囲】

【請求項1】 予め定めるプログラムに従って制御が行わ
れる電子制御装置において、

処理の順序によって結果が異なる予め定める複数の演算
処理を、プログラム実行中に処理が必ず行われる箇所に
挿入して、予め定める順序で実行する処理手段と、
処理手段の演算処理結果を予め定める期待値と比較し、
比較結果が不一致のとき異常状態であると判断する監視
手段とを含むことを特徴とする電子制御装置。

【発明の詳細な説明】

【0001】

【課題上の利用分野】 本発明は、電子制御装置、特に自
己診断機能を有する電子制御装置に関する。

【0002】

【従来の技術】 たとえば、自動車にはマイクロコンピュ
ータを用いた電子制御装置が搭載され、燃料噴射制御な
どのエンジン系制御や、サスペンション制御などの走行系
統の制御を行わせており、電子制御装置への依存度が高
まっている。したがって、それぞれの電子制御装置の動
作を監視し、確認することは、安全運転上、絶対に必要
不可欠であり、とりわけマイクロコンピュータを構成す
るCPUが、不測の原因により暴走することがある
ことが知られる。この場合、その暴走を即時に検出し、異常
が検知されなければならない。このため、CPUが実行
するプログラム中に、当該プログラムを所定の順序にお
り、即ち設定されたステップにしたがって実行したかと
うかを確認するための処理プログラムを、電子制御装置
本来の動作プログラムに付加し、CPUの動作を確認す
る手法が用いられる。このような所定のステップに対応
して挿入される動作チェックのための処理プログラム
を、一般にゲートと称する。

【0003】 図10は、プログラムとゲートの関係を示
すフローチャートである。一つのメインルーチンでは、処
理A、B、C、Dの4つの動作プログラムと、それぞれ
に対応して付けられた4個のゲートG1、G2、G3、
G4（総称するときはゲートGという）と、1個の判断
ゲートG5の5個のゲートプログラムから成り、処理A
～Dのプログラムが所定回数繰返されるものとする。そ
れぞれに挿入して示された4つのゲートA～Dと、ゲ
ートG1～G5は、いずれも枠内には当然に複数のステッ
プを含むものであるが、ここでは便宜上、1つの枠を1
つのステップと考える。図10から明らかのように、た
とえば電子制御装置本来の動作の1つである処理Aのプ
ログラムを実行した後、次の処理Bのプログラムに進む
ためには、その間に必ずゲートG2のステップを通過し
なければならない。即ち1つの処理が終わって次の処理
に移るために必ず通らなければならないステップ箇所に
ゲートGによる処理が設けられ、ゲートGのプログラム
により当該ゲートを通過したことを表す処理結果が記憶
され、後段でゲート通過数チェックされるようになっ

ている。

【0004】 ステップm1でメインルーチンに入ると、
ステップm2でゲートG1の処理により、RAMの所定
のエリアに「1」が書き込まれ、ゲートG1を通過した
ことが記憶される。ゲートG1を通過するとステップm
3に進み、処理Aのプログラムが実行される。次のステ
ップm4に進みゲートG2によって前記RAMの値に
「1」が加算され、「2」が記憶される。加算は前記A
LU3によって行われる。これによってゲートG2を通
過したことが記憶される。ステップm5で処理Bのプロ
グラムが実行されると、ステップm6でRAMの値にさ
らに「1」が加算され、「3」となる。次のステップm
7で処理Cが実行されると、ステップm8でゲートG4
による処理で、RAMには「4」が記憶される。ステッ
プm9で処理Dが実行された後ステップm10で、判断
ゲートG5によって前記RAMの値が、所定値である
「4」になっているかどうかチェックされる。「4」
であれば4個のゲートG1～G4を通過して正常と判断
されてステップm2へ戻り、「4」でなければ所定のゲ
ート数ではないため異常と判断されてステップm11へ
移って異常フラグがONされた後にステップm2へ戻
る。このようにして、ゲートGでの処理によって電子
制御装置の動作が所定のステップを通過したかどうか
が確認される。

【0005】

【発明が解決しようとする課題】 しかしながら、上述の
先行技術による電子制御装置でのチェック動作は、ゲー
ト通過ごとに1を加えて記憶するといった単純な処理に
過ぎず、このため順序どおりにステップを通過したかど
うかについての判断できない。たとえばなんらかの原因
で、ミステップして、処理A→C→D→Bと進んだ場
合でも、RAMには「4」が記憶されるので、正常と判
断されてしまう。また先行技術ではCPU内部のAL
U、レジスタ部などのチェックは行われていない。この
ため電子制御装置内のCPUの動作をチェックする上で
問題がある。

【0006】 本発明の目的は、前述の問題点を鑑みてな
されたものであって、電子制御装置の動作が正しい順序
で行われているかどうかをチェックすることができ、さ
らに演算処理のチェックもあわせて行えるようにするこ
とである。

【0007】

【課題を解決するための手段】 本発明は、予め定めるプ
ログラムに従って制御が行われる電子制御装置におい
て、処理の順序によって結果が異なる予め定める複数の
演算処理を、プログラム実行中に処理が必ず行われる箇
所に挿入して、予め定める順序で実行する処理手段と、
処理手段の演算処理結果を予め定める期待値と比較し、
比較結果が不一致のとき異常状態であると判断する監視
手段とを含むことを特徴とする電子制御装置である。

【0008】

【作用】本発明による電子制御装置は、処理の順序によって結果が異なる予め定められた複数の演算処理を、プログラム実行中に処理が実行される箇所に挿入して、予め定められた順序で実行する処理手段と、処理手段の演算処理結果を予め定められた期待値と比較し、比較結果が不一致のとき異常状態であると判断する監視手段とを含む。処理手段は処理の順序によって結果が異なる演算処理を行わせ、その演算結果と期待値とを、監視手段によって比較する。比較結果が不一致のときは、プログラムの実行順序または演算処理が正しく行われていないので、異常状態と判断する。

【0009】

【実施例】図1は、本発明の一実施例による電子制御装置の構成を示すブロック図である。マイクロコンピュータ1は、CPU2、ROM、RAMなどのメモリ部7、I/Oポート8など、集積化された各ブロックと、入力インターフェース9、出力インターフェース10および警告インターフェースなどの表示部11から構成され、エントリなどの制御対象に記憶された複数のメモリ12とフラッシュメモリ13が、ワイヤハーネス12、13を介して前記入力/出力インターフェース9、10にそれぞれ接続されている。CPU2と前記各ブロック間には、バスライブラリ14が配線され、センサ12-CPU2-プログラマチック13によるワイヤードバス制御が所定のプログラムに準じて実行される。プログラムの速度で動作と判断されると、CPU2は表示部11に対してプログラムを出し、メータ指示やランプ点灯などの警告動作を行わせる。

【0010】CPU2は、算術演算部3、レジスタ部4、RAM5、制御部6などから構成される。算術演算部(以下、ALUという)3は、図示しないアキュムレータやレジスタレジスタ部4により構成され、2進数による算術演算とAND、ORなどの論理演算を行う。演算速度のデータはレジスタ部4に入力され、演算結果はRAM5あるいはレジスタ部4の所定のメモリセルに記憶される。レジスタ部4には、アキュムレータAc、X、Y番地を指定して対応するメモリセルを呼び出すX、Yレジスタが含まれる。

【0011】図2は、図1に示す実施例のCPU2を構成する動作を説明するためのフローチャートである。メインルーチンとは、たとえば処理A、B、C、Dからなる4つのプログラムと、その間に挿入された4つのゲートG1、G2、G3、G4(総称するときはゲートGという)と、1つの制御ゲートG5の計5つのゲート処理プログラムを含む。本実施例では、処理手段である複数のゲートG1~4ごとに予め設定されたゲート処理プログラムによる演算処理を行い、その演算結果と予め定められた期待値とを、監視手段である判断ゲートG5により比較し、比較結果が不一致のとき異常と判断する

るのである。演算には前記ALU3とレジスタ部4を用いられる。

【0012】図1のブロック図をあわせて参照して説明する。ステップA1でメインルーチンの実行手順に入り、ステップA2でゲート1による処理が行われ、予め定められたコード「001B」(Bは2進表記を示す)がRAM5のうちのメモリセルには使用しない所定のメモリエリアで書き込まれ、ゲートG1を通過したことが記憶される。次のステップA3で処理Aのプログラムに基づき動作が実行されると、ステップA4に進んでゲートG2による処理が行われる。ここでは前記RAM5のメモリエリアの値(1001B)に「1」が加算される算術演算が予め設定されており、RAM5のメモリエリアの値は「1101B」に更新される。これによりゲートG2を通過したことが記憶される。

【0013】次のステップA5で処理Bのプログラムが実行されると、ステップA6に進んでゲートG3による処理が行われる。ここではRAM5のメモリエリアの値(1010B)が2倍される算術演算が予め設定されており、RAM5のメモリエリアの値は「0110B」に更新される。これによりゲートG3を通過したことが記憶される。次のステップA7で処理Cのプログラムが実行されると、ステップA8に進んでゲートG4による処理が行われる。ここでは予め定められた値0010BとRAM5のメモリエリアの値0110BとをORをとる論理演算が設定されており、RAM5のメモリエリアの値は「0110B」に更新される。これによりゲートG4を通過したことが記憶される。次のステップA9で、処理Dのプログラムが実行されると、ステップA10に進み、判断ゲートG5による処理が行われる。これまで述べたように、ステップを順次おりに通過すれば、RAM5のメモリエリアには期待値と同一の「0110B」が記憶されている筈であり、期待値とRAM5のメモリエリアの値とが一致すればステップA2へ戻る。不一致の場合は、メインルーチンが正しい順序で実行されなかった異常状態と判断され、ステップA11へ移って異常フラグがONされた後、ステップA2へ戻る。なお、異常フラグとしては、RAM5の中の特定のメモリセルを使用する。

【0014】このように本実施例では、ゲートG1~4ごとに、定数と、算術演算と、論理演算の順序を予め定めて期待値を設定し、一方ステップが進行して最終ゲートG5で、得られた値を前記期待値と比較することにより、正常か異常かを判断する。したがって先行技術のように、単に1を加算して判断するのでは異なり、一つでもプログラムの実行順序が異なれば、絶対に期待値とは一致せず、たまたま異常が発見されるのである。これによりプログラムが所定のステップで動作しており実行されたか否かを的確にチェックでき、プログラムの暴走などを速やかに検知することができ

【0015】本実施例では、前記演算を行うALU3の機能が正常かどうかを判断するチェック機能を備えている。図3と図4は、その動作を示すフローチャートである。ここではゲートG5を繰り返して何回か通過するプログラムが含まれるメインルーチンを選定している。プログラムが進行し、ステップP1でゲートG1に入ると、制御部6は今回のゲートG1通過が何回目であるかを判断する。通過回数の順にはたとえば図示しないカウンタが用いられる。ゲートG1の通過が1回目であれば、ステップP3に進み、2回目であればステップP5に、3回目であればステップP7に進む。それ以上の回数についても同様である。通過回数に対してステップP3、P5、P7、...での演算手段が予め定められており、たとえば1回目であるステップP3では、RAM5内に予め定められたエリアRの値に「1」が加算される。2回目であるステップP5では、Rの値が2倍される。3回目であるステップP7ではRの値とたとえば定数「1001B」とのANDがとられる。このように回数に対応する演算手段とその結果に基づいて更新されたRの値

(1, 2, 3, ...)が記憶され、ステップP8で次のプログラム、たとえば処理Aに進む。以下、ゲートG2, G3, ..., ごとに同様の演算が行われ、通過回数1, 2, 3, ...に対応する最終の演算結果である値R1, R2, R3, ...が、次に述べるゲートG5で期待値Nと比較される。これを図4で説明する。前述したように、ゲートG1~4を通過すること、RAM5のRの値が更新され、またゲートG1~4の通過回数がかつた値などにより記憶されている。ステップP1でゲートG5に入ると、ステップP2で制御部6はゲートG5の通過回数を調べ、1回目であればステップP3に進み、2回目であればステップP4からステップP7に、3回目であればステップP6からステップP7にそれぞれ進む。ゲートG5の通過回数に対応する期待値N1, N2, N3, ...が予め設定されており、ステップP3, P4, P5, ...で制御部6は、前記した値Rと前記期待値N1, N2, N3, ...とを比較する。両者が一致すれば

ステップP9で次のプログラムに進む。不一致であればステップP48に移って異常フラグがONされる。その他の回数に対応するステップについても同様である。ここではRAM5のRの値(1, 2, 3, ...)は、ALU3の演算結果であり、これと期待値N1, N2, N3, ...とを比較することによって、ALU3の機能の良否がチェックされ、異常フラグがONされれば、ALU3に異常が生じていることが検知される。このように本実施例では、CPUのステップ動作の良否だけでなく、CPU2を構成するALU3の機能の良否もチェックすることができ

【0016】本実施例は、またレジスタ部4をチェックする機能を備えている。レジスタ部4には、図5に示されるXおよびYレジスタが含まれる。CPU2には、レ

ジスタX、Yを用いてメモリをアクセスする命令が備えられている。Xを1, 2, 3, 4と順次変化させ、Yは0001B, 0010B, 0100B, 1000Bのようになじり、0010Bのうしろの桁を「1」とした1回の番地指定すれば、対応するメモリセルM(1, 1), M(2, 2), M(3, 3), M(4, 4)が選択される。たとえばX=1, Y=0001BならばメモリセルM(1, 1)が呼び出される。したがってXを前記のように変化させ、Yのどれかの桁を1にして番地指定すれば、4つのゲートG1~4を通過する順に異なるメモリセルを特定できる。

【0017】次にレジスタ部4をチェックする動作について説明する。図6~図9は、その動作を説明するためのフローチャートであり、図6はゲートG1を、図7はゲートG2を、図8はゲートG3を、それぞれ通過する際の動作を示している。メモリセルはチェック動作開始の際にはクリアされる。

【0018】図6を参照して、ステップP1でゲートG1に入り、ステップP2でアキュムレータAcの内容が0クリアされ、ステップP3でアキュムレータAcに1が加えられる。次にステップP4, P5で前記X, Yレジスタに対してX=1, Y=1を設定し、ステップP6で前記アキュムレータAcからメモリセルM(X, Y)に内容が転送される。転送が終われば、ステップP7でゲートG1から処理Aなどの次のステップ動作に移行する。動作が正常なら、メモリセルM(1, 1)には「1」が記憶されている。

【0019】図7はゲートG2での動作を示す。ステップP1でゲートG2に入り、ステップP2, P3でX=1, Y=1と設定され、ステップP4でX、Y番地の内容がアキュムレータAcに転送される。ステップP5でアキュムレータAcに1が加えられる。次にステップP6, P7でX=2, Y=2と設定され、ステップP8でアキュムレータAcからX、Yで指定されるメモリセルに内容が転送される。転送が終わればステップP7でゲートG1から処理Bなどの次のステップ動作に移行する。動作が正常なら、メモリセルM(2, 2)には「2」が記憶されている。

【0020】図8はゲートG3での動作を示す。ステップP1でゲートG3に入り、ステップP2, P3でX=2, Y=2と設定され、ステップP4でX、Yで指定される番地の内容がアキュムレータAcに転送される。ステップP5でアキュムレータAcに1が加えられる。次にステップP6, P7でX=3, Y=4と設定され、ステップP8でアキュムレータAcからメモリセルに内容が転送される。転送が終わればステップP7でゲートG3から処理Cなどの次のステップ動作に移行する。動作が正常ならメモリセルM(3, 4)には「3」が記憶されている。なお次のゲートG4を通過する際の動作は、前述までの動作と同様であって、ここでは動作が正

(7)

特開平06-149626

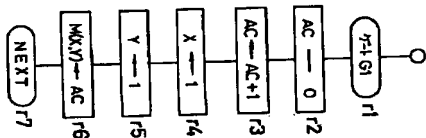
(8)

特開平06-149626

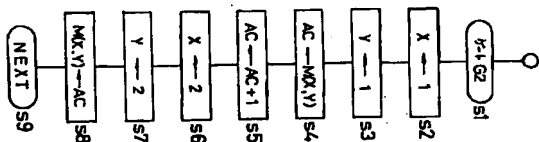
【図5】

X\Y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2																
3																
4																

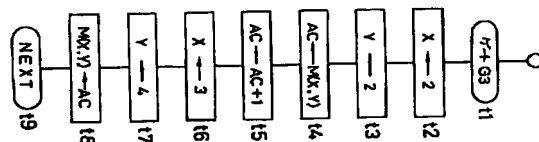
【図6】



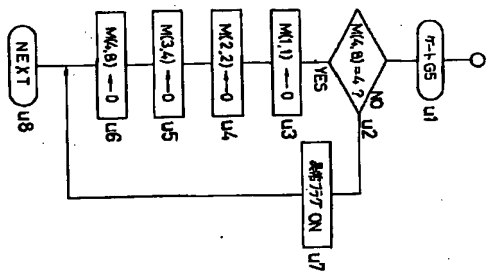
【図7】



【図8】



【図9】



【図10】

